

MODÜL 4

KOŞULLU VE MANTIKSAL İFADELER



Şekil 4.1: Bölümle ilgili örnek uygulamalara karekoddan ulaşabilirsiniz.

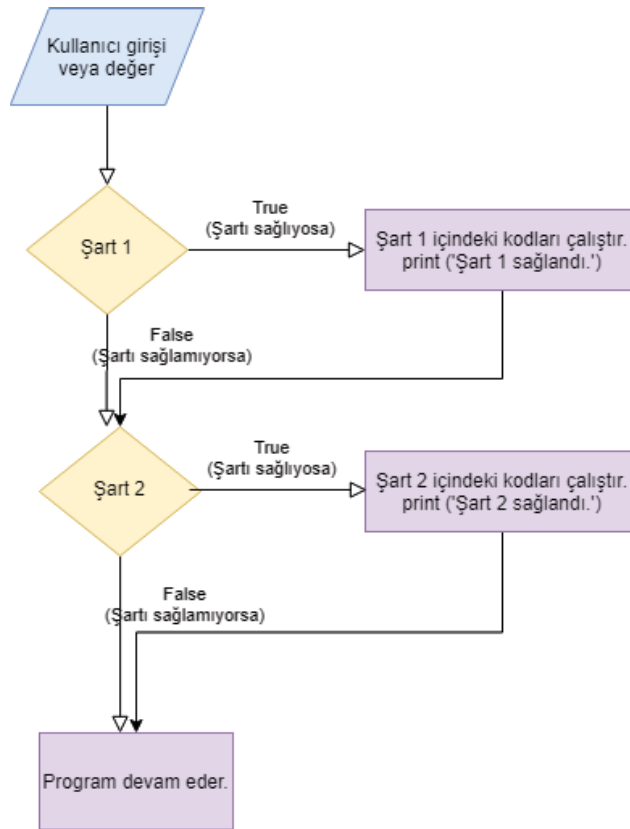
4.1. Koşullu İfadeler

Program yazarken kodlar sıralı olarak alt alta satırlar şeklinde yazılır. Programın akışında bazı dallanmalar (farklı durumlar için farklı kodların çalışmasını istediğimiz durumlar) olabilir. Programlarda belirli koşul veya durumlar için çalışması istenen kodlar koşullu ifade blokları kullanılarak oluşturulur.

Örneğin, sadece üyelerin giriş yapabileceği bir program hazırlarken kullanıcı adı ve parolası doğru olan kişilerin sisteme erişimine izin veren kodlar yazılır. Bunun için programa koşul ifadeleri eklenmesi gerekir. Koşul sağlanıyorsa menülere erişim izni verecek kodlar çalışmaya başlar. Koşul sağlanmıyorsa uyarı mesajı verilerek sisteme giriş izni verilmez.

MODÜL 4

Mantıksal operatörler sonuç olarak “boolean” veri tipinde değer verir. Eğer koşul sağlanırsa “True” değeri döndürürken koşul sağlanmazsa “False” değeri döndürür. “Boolean” veri tipi bu iki değerden başka bir değer alamaz. Bu durum koşullu ifadeler üretme olanağı sağlar. Koşullu ifadelerin sonucu “boolean” değer kontrol edilerek program akışı yönlendirilebilir. Koşul ifadesi ve “True-False” akışı Şekil 4.2’de daha iyi görülebilir.



Şekil 4.2: Koşullu ifadeler

Şekil 4.2’de görüldüğü üzere kullanıcıdan alınan veri “Şart 1” yapısına geldiğinde şartı sağlıyorsa (True) bu kod girintisi (blok) içindeki komutlar çalışır. “Şart 1” yapısında şart sağlanmıyorsa “blok” atlanarak sonraki kodlara geçilir. Sonrasında yeni şart yapısı (Şart 2) aynı şekilde kontrol edilir ve akış devam eder.

4.2. Mantıksal İfadeler ve Bağlaçlar

Bir mantıksal ifadeyi diğer mantıksal ifadelerle bağlamanın farklı yolları vardır. Modül 2’de açıklanan ilişkisel operatörler ve mantıksal bağlaçlar kullanılarak (or, and, not vb. gibi) farklı koşul durumları oluşturulabilir.

kullaniciAdi=='Admin' and kullanıcıParola=='123456' Kullanıcı adı ve parola doğru girilirse “True” değeri döndürür.

Örnek

1

Aşağıdaki örnek kodda kullanıcı adı ve şifresi doğru girilirse ekrana “True” ikisinden biri bile yanlış olursa “False” değeri döndürür.

```
kullaniciAdi=input('Kullanıcı Adı:')
kullaniciParola=input('Parola:')
print(kullaniciAdi=='Admin' and kullanıcıParola=='123456')
Kullanıcı Adı:Admin
Parola:123456
True
```

Örnek

2

Bölümü Bilgisayar veya Elektronik olanları seçmek için bir kod yazalım. bolum=='Bilgisayar' “or” bolum=='Elektronik' koşullarından biri doğruysa “True” değilse “False” değeri döndürür.

```
bolum=input('Bölümünüzü giriniz: ')
#Büyük harf küçük harf duyarlılığını ulatmayın
print(bolum=='Bilgisayar' or bolum=='Elektronik')
Bölümünüzü giriniz: Bilgisayar
True
```

Örneklerde sadece mantıksal operatörlerin sonucu “boolean” değeri ekrana yazdırıldı. Mantıksal operatörler koşul ifadeleriyle birlikte kullanıldığında belirli şartlarda belirli kod blokları çalıştırılabilir.

4.3. Python Blok Yapısı

Python’da (başka programlama dillerinde de) kodlar belirli alt kümeler hâlinde (blok) ifade edilir. Bu yapı Python’da girinti (Indentation) ile oluşturulur. Python’da dikey olarak aynı hizadaki kodlar aynı blok yapısında yer alır. Kod bloklarının kolaylıkla ayırt edilebilmesi için bir sekme (4 karakter boşluk) kullanılması önerilmektedir. Eğer bir IDE (bütünleşik program geliştirme ortamı) kullanılıyorsa bu girintiler otomatik olarak ayarlanacaktır. Kullanılan geliştirme ortamında ayarlar bölümünde bir girintinin kaç karakter olacağını belirlediği ayarlar bulunmaktadır. Bir karakter boşluk ile oluşturulan girinti bile yeni blok yapısını oluşturur. Bir kod bloğu kendi içinde tutarlı bir yapıdır. Döngüler, fonksiyonlar ve koşul ifadeleri kod blokları kullanılarak oluşturulur.

```
[ ] 1  yasi=17
      2  adi='Tahsin'
      3  if (yasi>=18):                1. Blok
      4      print('1. Şart sağlandı')    if Bloğu
      5      print ('1. if bloğunun içindesiniz')
      6  if (adi=='Tahsin'):            2. Blok
      7      print('2. Şart sağlandı')    if bloğu
      8      print ('2. if bloğunun içindesiniz')
      9  print('Normal program akışı girinti yok')
```

Şekil 4.3: Python blok yapısı

Şekil 4.3’te “if” ifadesinin altında girinti oluşturularak bir blok (1. Blok) oluşturulmuştur. “yasi >= 18” mantıksal ifadesinin sonucu blok içindeki kodların tamamının atlanmasına veya çalıştırılmasına neden olur. Şekilde iki blok yapısı görünmektedir. Python’da kod yazarken girintilere dikkat etmek gerekir. Girintiler blok yapısını belirlediği için programın yanlış çalışmasına veya çalışmamasına neden olabilir.

Örnek

3

Aşağıda herhangi bir karar yapısı, döngü veya fonksiyon olmadan ikinci satırdaki kodu girintili yazıldığı için hata verir.

```
print('Blok yapısı')
    print('Girinti')
File "<ipython-input-7-b54d1a37c7f1>", line 2
    print('Girinti')
    ^
IndentationError: unexpected indent
```

4.4. if yapısı

Bu yapıda, belirli komutların çalışması, bir koşula bağlıdır. Koşul sağlanmazsa herhangi bir işlem yapılmaz.

Kullanımı: Aşağıda bir “if” bloğu gösterilmektedir “if” bloğunun dikey hizasının sağında olan kod satırları koşul gerçekleştiğinde çalışır. Bu kodlar “if” bloğunda yer almaktadır. Büyük eşittir operatörü karşılaştırma sonucu “boolean” (True veya False) bir değer verir. True değer verirse “if” bloğu içinde (girintide olan) kodlar çalışır. Koşul sağlanmazsa yani “False” değeri verirse bloğun içine girilmez bloktaki kodlar atlanır.

Örnek

4

Kullanıcının yaş değerini alarak 18’e eşit veya büyük olması hâlinde ona mesaj veren kod:

```
yasi=int(input('Lütfen yaşınızı giriniz: '))
if (yasi>=18):
    print('Oy kullanabilirsiniz.')
print ('Program bitti.')
Lütfen yaşınızı giriniz: 18
Oy kullanabilirsiniz.
Program bitti.
```

MODÜL 4

“if” bloğunun içindeki kod ancak şart sağlandığında çalışır ve blok bittikten sonra program akışı devam eder. Şart sağlanmazsa blok atlanır. Python alt satırdaki kodları yorumlar ve ona göre işlem yapar. Aşağıdaki örnek 18’den küçük bir yaş girilerek çalıştırılmıştır.

```
yasi=int(input('Lütfen yaşıınızı giriniz: '))
if (yasi>=18):
    print('Oy kullanabilirsiniz.')
print ('Program bitti.')
Lütfen yaşıınızı giriniz: 15
Program bitti.
```

Örnek

5

“adi” ve “yasi” değişkenlerinin değerlerine göre hangi blokların çalıştığına dikkat ediniz.

```
yasi=17
adi='Tahsin'
if (yasi>=18):
    print('1. Şart sağlandı.')
    print ('1. if bloğunun içindesiniz.')
if (adi=='Tahsin'):
    print('2. Şart sağlandı')
    print ('2. if bloğunun içindesiniz.')
print('Normal program akışı girinti yok.')
2. Şart sağlandı.
2. if bloğunun içindesiniz.
Normal program akışı girinti yok.
```

Koşul ifadelerinde birden fazla koşul birlikte kullanılabilir. Bunun için operatörler konusunda açıklanan mantıksal operatörler kullanılır. Örnekte “and” operatörü ile iki koşulun birlikte sağlanma şartı koşulmuştur.

Örnek**6**

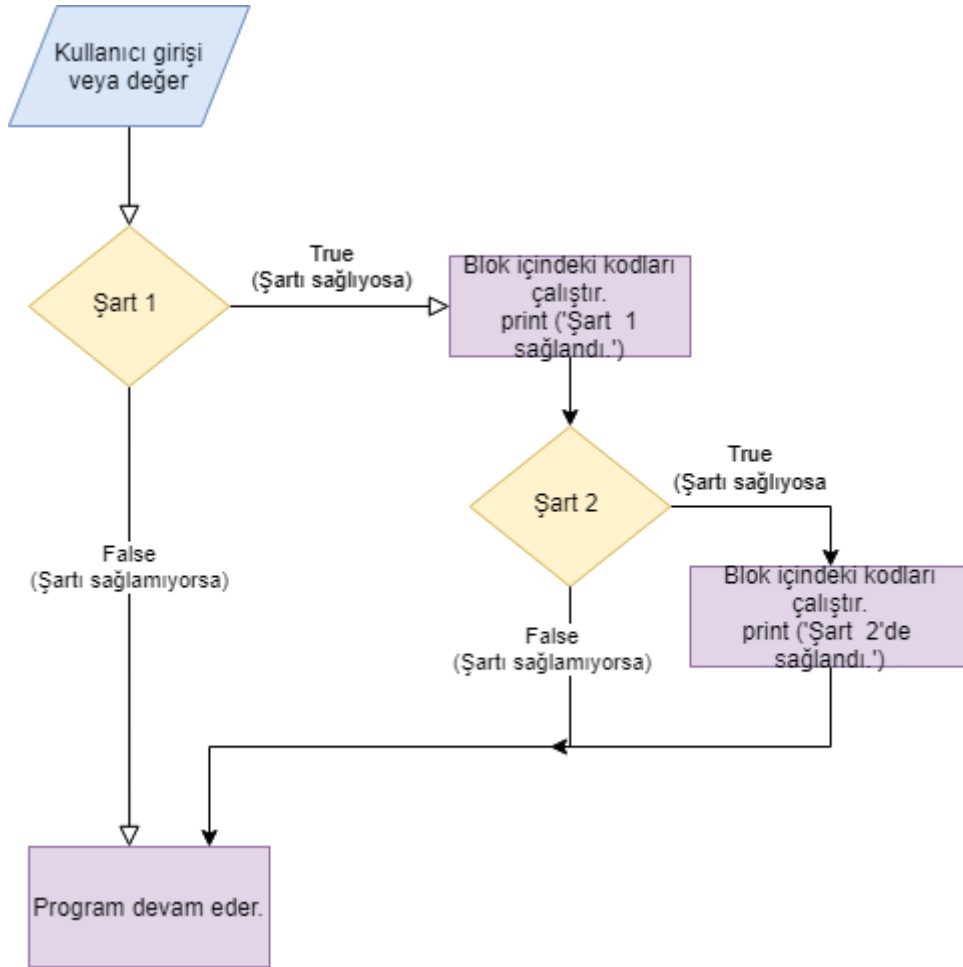
Kullanıcının girdiği kullanıcı adı ve parolayı kontrol ediniz.

```
kullaniciAdi=input('Kullanıcı Adı:')
kullaniciParola=input('Parola:')
if (kullaniciAdi=='Admin' and kullaniciParola=='123456'):
    print('Giriş başarılı.')
    print ('Menülere erişebilirsiniz.')
Kullanıcı Adı:Admin
Parola:123456
Giriş başarılı.
Menülere erişebilirsiniz.
```

Örnekte şartlardan biri bile sağlanmazsa “if” bloğundaki kodlar çalışmaz.

4.5. İç İçe Koşul İfadeleri

Yukarıdaki örnekte birden fazla koşulu “and” operatörünü kullanarak kontrol edilmiştir. Aynı işlem, iç içe koşul ifadeleri kullanarak da yapılabilir.



Şekil 4.4: İç içe koşul ifadeleri

Şekil 4.4’te görülen koşullu ifadelerden birincisinin yani ilk “if” bloğundaki şart sağlanırsa (Şart 1) o bloktaki kodların çalıştığı bilinmektedir. Bu şart ifadesinin içine bir koşul ifadesi yani ikinci bir “if” bloğu (Şart 2) daha eklenebilir. Şart 2’deki koşul da sağlanırsa bu kez “if” bloğunun içindeki ikinci “if”

bloğundaki kodlar da çalışır. İkinci koşul ifadesindeki şart sağlandığında birinci koşul ifadesindeki şart zaten sağlanmış olacağından (sağlanmasaydı ikinci “if” bloğu kodu birincinin içinde olduğu için zaten çalışmazdı) iki koşul ifadesi de (Şart 1 and Şart 2) sağlanmış olur. Koşul ifadelerinden birincisi sağlanır ikincisi sağlanamazsa ikinci “if” bloğundaki kodlar atlanır. Bu kullanım örnekteki “and” operatörü kullanımına benzetmekle birlikte koşullardan sadece birincisinin sağlandığı durumlar için kodlar oluşturmaya olanak verir. İç içe koşul ifadelerinin sayısı (iç içe 3 koşul gibi) ihtiyaca göre artırılabilir.

Örnek**7**

Kolay anlaşılması için verilen örnekteki ikinci “if” bloğu içindeki kodların giritisine dikkat ediniz.

```
kullaniciAdi=input('Kullanıcı Adı:')
kullaniciParola=input('Parola:')
if (kullaniciAdi=='Admin'):
    print('Kullanıcı adı doğru')
    if (kullaniciParola=='123456'):
        print('Giriş başarılı.')
        print ('Menülere erişebilirsiniz.')
Kullanıcı Adı:Admin
Parola:123456
Giriş başarılı.
Menülere erişebilirsiniz.
```

İlk şart yapısında kullanıcı adının doğru olup olmadığı kontrol edilmektedir. Eğer şart doğruysa içteki 2. şart bloğu çalışarak parola kontrolü yapacaktır. İki şart sağlanırsa ekrana tüm mesajlar yazdırılacaktır. Sadece 1. şart sağlanırsa kullanıcı adı doğru girilip parola yanlış girilirse “Kullanıcı adı doğru” mesajı ekranda görünecektir. Kullanıcı adı yanlış girilirse hiçbir mesaj görünmeyecektir.

4.6. if-else Yapısı

“if” yapısında şart sağlanırsa blok içindeki kodlar çalışmaktadır. Ancak şartın sağlanmadığı durumlarda herhangi bir işlem yapılmaz. “else” ifadesi değilse anlamındadır. Yani şartın sağlanmadığı durumda çalışacak kodlar “else” bloğuna yazılır.

Kullanımı: “else” bloğu da if bloğu gibi ayrı bir blok olarak yazılır. Bir “if” bloğundan sonra gelen else bloğu aynı girinti seviyesinde olmalıdır. “else” bloğu “if” ile birlikte kullanılır.

Örnek

8

Aşağıdaki örnekte kullanıcının girdiği sayının çift - tek olduğunu bulan bir kod yazılmıştır.

Çift bir sayı girildiğinde “if” bloğunun içindeki kodlar çalışır.

```
sayi1=int (input ('Lütfen bir sayı giriniz: '))
if ((sayi1%2)==0):
    print('Girdiğiniz sayı çifttir.')
else:
    print('Girdiğiniz sayı tektir: ')
Lütfen bir sayı giriniz: 12
Girdiğiniz sayı çifttir.
```

Tek bir sayı girildiğinde “else” bloğunun içindeki kodlar çalışır.

```
sayi1=int (input ('Lütfen bir sayı giriniz: '))
if ((sayi1%2)==0):
    print('Girdiğiniz sayı çifttir.')
else:
    print('Girdiğiniz sayı tektir: ')
Lütfen bir sayı giriniz: 13
Girdiğiniz sayı tektir:
```

Örnek

9

Koşullu ifade, operatörler ve bağlaçlarla daha etkili yapılabilir. Yukarıdaki kullanıcı adı ve parolası örneğinde kullanıcının girdiği kullanıcı adı ve parola bu örnekte “and” ile birlikte kontrol edilmiştir.

```
kullaniciAdi=input('Kullanıcı Adı:')
kullaniciParola=input('Parola:')
if (kullaniciAdi=='Admin' and kullaniciParola=='123456'):
    print('Giriş başarılı.')
    print ('Menülere erişebilirsiniz.')
else:
    print ('Yanlış kullanıcı adı veya şifre')
Kullanıcı Adı:Admin
Parola:123456
Giriş başarılı.
Menülere erişebilirsiniz.
```

Örnek

10

“if else” yapısına ilişkin başka bir örnek aşağıda verilmiştir. Bu örnekte şartlar “and” bağlacıyla birleştirilmiştir.

```
yasi=int(input('Lütfen yaşınızı giriniz: '))
bolum='Bilgisayar'
yabanciDil=True
if (yasi>=18 and yasi<35 and bolum=='Bilgisayar' and yabanciDil==True):
    #Aşağıdaki kodun çalışması için yukarıdaki 3 şartın da sağlanması gerekir.
    print('Mülakata katılabiliyorsunuz.')
else:
    print('Şartlarınız tutmuyor.')
Lütfen yaşınızı giriniz: 20
Mülakata katılabiliyorsunuz.
```

MODÜL 4

4.7. if-elif-else Yapısı

Bu yapıda koşullar art arda verilir. if ile verilen koşulun devamında 'değilse şu ise' anlamına gelen "elif" ifadesi yer alır. Yapının en sonunda ise 'hiçbiri değilse' anlamında else ifadesi yer almaktadır. Her ifade kendi bloğundaki kodları çalıştırır. "if, elif ve else" bloklarının girinti düzeyleri aynı olmalıdır.

Her koşul ifadesi bir "if" bloğu formatında yazılabilirdi ancak bu durumda program akışında tüm koşul ifadeleri tek tek kontrol edilirdi. "if-elif-else" yapısında ise şart sağlandığında veya else ifadesine gelindiğinde ilgili bloktaki kodlar çalışır ve tüm "if-elif-else" bloğundan çıkılır. Birbirleriyle bağlantısı olmayan koşullar ayrı "if" blokları şeklinde verilebilir. Ama bir değerin belirli aralıktaki şartları sağlayıp sağlamadığı kontrol edilirken "if-elif-else" yapısını kullanmak daha uygundur. Bu yapıda koşullardan biri sağlanıyorsa diğer koşullar kontrol edilmez. Alınan değer "if-elif-else" yapısındaki yalnız bir koşulu sağlayabilir.

Örnek

11

Bir kullanıcının sınav puanını alarak durumunun değerlendirilmesi:

```
sinavPuani=int(input('Puanınız giriniz (0-100): '))
if sinavPuani>=85:
    print('Pek iyi')
elif sinavPuani>=70:
    print('İyi')
elif sinavPuani>=55:
    print('Orta')
elif sinavPuani>=45:
    print('geçer')
else: print('Kaldı')
Puanınız giriniz (0-100): 65
Orta
```

Örnek

12

Öğrencilerin başarı puanlarını hesaplayan ve sonucu değerlendirerek harf notu olarak döndüren bir program yazımı:

```
basariPuani=-1#kontrol değerimiz
vizePuani=int(input('Vize puanını giriniz: '))
finalPuani=int(input('Final puanını giriniz: '))
vizeOraniYuzde=(int(input('Vize oranını % olarak giriniz (30, 40 gibi): ')))
finalOraniYuzde=(int(input('Final oranını % olarak giriniz (70, 60 gibi): ')))
#Alınan verilerin kontrol edilmesi
if (vizePuani>100 or vizePuani<0 or finalPuani>100 or finalPuani<0 ):
    print ('Girdiğiniz vize veya final puanlarınızı kontrol ediniz.')
elif (vizeOraniYuzde+finalOraniYuzde)<100:
    print ('Girdiğiniz vize veya % oranlarınızı kontrol ediniz toplam 100 olmalıdır.')
else:
    basariPuani=(vizePuani* vizeOraniYuzde/100)+(finalPuani* finalOraniYuzde/100)
    #Bir hata yoksa basariPuani hesaplanır.
    #Hesaplanmazsa ilk verilen -1 değeri kalır.
if (basariPuani>=0):
    print('Başarı puanı: ', basariPuani )
    if (basariPuani>= 80 and basariPuani <= 100):
        print('Başarı notu : A')
    elif (basariPuani>= 70 and basariPuani <80):
        print('Başarı notu : B')
    elif (basariPuani>= 60 and basariPuani <70):
        print('Başarı notu : C')
    elif (basariPuani>= 50 and basariPuani <60):
        print('Başarı notu : D')
    elif (basariPuani< 50 ):
        print('Başarı notu : F')
Vize puanını giriniz: 70
Final puanını giriniz: 80
Vize oranını % olarak giriniz (30, 40 gibi): 30
Final oranını % olarak giriniz (70, 60 gibi): 70
Başarı puanı: 77.0
Başarı notu : B
```

4.8. Bölüm Sonu Örnekleri

1. Kullanıcının girdiği tam sayının “Negatif”, “Pozitif” ya da “Sıfır” olduğunu yazdıran programın kodlarını yazınız.
2. Öğrencinin sınav ortalamalarını kullanıcıdan alan ortalama en az 50 ise geçti değilse kaldı yazan programı yapınız.
3. Kullanıcın girdiği iki sayıyı karşılaştırarak sayı sayıdan büyüktür, küçüktür veya sayılar eşittir mesajı veren kodları sadece “if” koşul yapısını kullanarak yazınız.
4. Yukarıdaki programı “if-elif-else” koşul yapısını kullanarak yazınız.

5. Aşağıdaki kodun çıktısı ne olur?

```
sayi1=12
sayi2=60
toplam=0
if sayi1<=sayi2:
    if sayi1%2==0:
        sayi1=sayi2
        toplam=sayi1+sayi2
    else: toplam=sayi2-sayi1
toplam+=toplam
print(toplam)
```

6. Yukarıdaki kodda sayi1=40 sayi2=13 değerleri için kodun çıktısı kaç olur?
7. Kullanıcıdan (1-4) arasında sayı alınacak, bu sayıya göre sırasıyla İlkbahar-yaz-sonbahar-kış yazan programın kodlarını yazınız.

8. Kullanıcıdan alınan dört kenar uzunluğuna göre şeklin kare, dikdörtgen veya diğer dörtgenlerden olduğunu belirten kodu yazınız.
9. Kenar uzunlukları girilen üçgenin çeşidini bulan programın kodlarını yazınız.
10. Beden kitle endeksini $\text{kilo}/(\text{boy}^2)$ formülü ile hesaplanarak bireyin kilo durumunu kontrol eden programın kodlarını aşağıdaki aralık durumlarına göre yazınız.
 Kitle Endeksi (KE) < 18.5 ise Zayıf,
 18.5 < (KE) <=25 ise Normal,
 25 < (KE) <= 30 ise Kilolu, (KE) > 25 ise birey obez sınıfına girmektedir.

Cevaplar

1.

```
sayi = input('Sayı : ')
if(int(sayi)<0):
    print("Sayı Negatif")
elif(int(sayi)>0):
    print("Sayı Pozitif")
else:
    print("Sayı Sıfır")
Sayı : -30
Sayı Negatif
```
2.

```
ort = input('Ortalamanızı Girin : ')
if(int(ort)>=50):
    print("Geçtiniz")
else:
    print("Kaldınız")
Ortalamanızı Girin : 60
Geçtiniz
```

MODÜL 4

```
3. sayi1= int(input('1. sayıyı giriniz: '))
sayi2= int(input('2. sayıyı giriniz: '))
if sayi1>sayi2:
    print('1. sayı 2. sayıdan büyüktür.')
if sayi1<sayi2:
    print ('1.sayı 2. sayıdan küçüktür.')
if sayi1==sayi2:
    print ('Sayılar eşittir.')
1. sayıyı giriniz: 12
2. sayıyı giriniz: 24
1.sayı 2. sayıdan küçüktür.
```

```
4. sayi1= int(input('1. sayıyı giriniz: '))
sayi2= int(input('2. sayıyı giriniz: '))
if sayi1>sayi2:
    print('1. sayı 2. sayıdan büyüktür.')
elif sayi1<sayi2:
    print ('1.sayı 2. sayıdan küçüktür.')
else: print ('Sayılar eşittir.')
1. sayıyı giriniz: 12
2. sayıyı giriniz: 24
1.sayı 2. sayıdan küçüktür
```

5. 240

6. 1


```
7. a=int(input("Mevsim No:"))
if(a==1):
    print("İlkbahar")
elif(a==2):
    print("Yaz")
elif(a==3):
    print("Sonbahar")
elif(a==4):
    print("Kış")
else: print("Aralıkta olmayan bir değer girdiniz")
Mevsim No: 2
Yaz
```

```
8. a=int(input("1. kenar:"))
b=int(input("2. kenar:"))
c=int(input("3. kenar:"))
d=int(input("4. kenar:"))
if(a==b==c==d):
    print("Kare!")
elif(a==c and b==d or a==b and c==d ):
    print("Dikdörtgen")
else: print("Diğer Dörtgen")
1. kenar:6
2. kenar:7
3. kenar:8
4. kenar:9
Diğer Dörtgen
```

MODÜL 4

```
9. a=int(input("1. kenar:"))
b=int(input("2. kenar:"))
c=int(input("3. kenar:"))
if(a!=b and a!=c and b!=c):
    print("Çeşitkenar Üçgen!")
elif(a==b==c):
    print("Eşkenar Üçgen!")
else: print("İkizkenar Üçgen")
1. kenar:3
2. kenar:5
3. kenar:6
Çeşitkenar Üçgen!
```



```
10. boy = float(input("Boy: Örnek 1.73----:"))
kilo = float(input("Kilo: Örnek: 78.40----:"))
endeks = kilo/(boy**2)
if endeks<18.5:
    print("Zayıfsınız")
elif endeks > 18.5 and endeks <=25 :
    print("Normalsiniz")
elif endeks > 25 and endeks <=30:
    print("Kilolusunuz")
elif endeks > 30:
    print("Dikkat! obez")
Boy: Örnek 1.73----:1.75
Kilo: Örnek: 78.40----:95
Dikkat! obez
```